the information received from the client 18 via the Internet 12 after decryption using the decryption engine 88. However, the decrypted data 114 may also comprise information in an unencrypted format intended to be transmitted from the server 20 to the client 18.

In operation, the server 20 receives the character string 54 from the client 18 and stores the character string 54 in the database 100 as a character string 116. Using the identification key 60 received from the client 18, the processor 80 accesses the relational data 102 of the database 100 to determine the private key 110 corresponding to the identification key 60. For example, as described above, the relational data 102 may comprise a look-up table relating each identification key 108 to a private key 110. Using the identification key 60, a corresponding identification key 108 may be identified, thereby also identifying the corresponding private key 110. The hashing engine 86 generates and stores a hash key 118 in the database 100 by hashing the private key 110 with the character string 116. The decryption engine 88 then decrypts the encrypted data 112 using the hash key 118. The decrypted data 114 is then stored in the database 100.

To authenticate the transmitted data and the identity of the client 18, the signature engine 90 is used to verify or authenticate the signature 56 received from the client 18. In operation, the signature engine 90 hashes the hash key 118 with the decrypted data 114 to generate a signature 120 which is stored in the database 100. The signature 120 may then be compared with the signature 56 to verify and authenticate the transmitted data and the identity of the client 18. If the signature 120 does not match the signature 56, the processor 80 may be configured to generate an alert or alarm to a user of the system 10 and/or discard the transmission data 104.

The present invention may also be used to transmit data from the server 20 to the client 18 via the Internet 12. For example, the string generator 84 may be used to randomly generate and store a character string 116 in the database 100. The hashing engine 86 may hash the private key 110 corresponding to the client 18 with the character string 116 to generate the hash key 118. Using the hash key 118, the engine 88 may be used to encrypt data to be transmitted to the client 18. The encrypted data and the character string 116 are then transmitted from the server 20 to the client 18 via the Internet 12. The client 18 may then decrypt the data using the character string 116

and the private key 62 similar to as described above in connection with the server 20. For example, the hashing engine 42 may be used to hash the character string 116 generated by the generator 84 with the private key 62 to generate the hash key 64 for decrypting the received encrypted data 112. The signature engine 90 may also be used to generate a signature 120 corresponding to the transmitted data by hashing the hash key 118 with the data similar to as described above in connection with the client 18. The signature 120 may then be transmitted to the client 18 via the Internet 12. The client 118 may then compare the signature 120 to a signature generated by the signature generator 46 using the hash key 64 and the decrypted data. The processors 30 and 80 may also be configured to incorporate a sequence number or identifier into the data 70 and 114 such that duplicate or out-of-sequence data transmissions received by either the client 18 or server 20 are discarded or rejected.

FIGURE 2 is a flowchart illustrating a method for secure data transmission in accordance with an embodiment of the present invention. The method begins at step 200, where the identification key 60 is stored in the database 50. At step 204, the private key 62 corresponding to the client 18 is also stored in the database 50. The client 18 receives data to be transmitted to the server 20 via the Internet 12 at step 206. At step 208, the string generator 40 generates a random character string 54 and stores the character string 54 in the database 50. At step 210, the hashing engine 42 generates the hash key 64 by hashing the private key 62 with the character string 54. At step 212, the encryption engine 44 encrypts the data 70 to be transmitted to the server 20 using the hash key 64 as an encryption password.

At step 214, the signature generator 46 generates the signature 56 by hashing the hash key 64 with the data 70. The character string 54, the encrypted data 72, the identification key 60 corresponding to the client 18, and the signature 56 are then transmitted to the server 20 via the Internet 12 at step 216.

FIGURE 3 is a flowchart illustrating a method for secure data transmission in accordance with another embodiment of the present invention. The method begins at step 300 where identification keys 108 corresponding to each client 18 are stored in the database 100. At step 302, private keys 110 relating to each of the identification keys 108 are also stored in the database 100. At step 304, the server 20 receives the

character string 54, the encrypted data 72, the identification key 60 corresponding to the transmitting client 18, and the signature 56 from the client 18.

At step 306, the processor 80 accesses the relational data 102 of the database 100. At step 308, the received identification key 60 of the client 18 is used to determine the private key 110 corresponding to the client 18. At step 310, the hashing engine 86 generates the hash key 118 by hashing the private key 110 with the character string 54 received from the client 18. At step 312, the decryption engine 88 decrypts the encrypted data 72 received from the client 18 using the hash key 118 and stores the decrypted data 114 in the database 100. At step 314, the signature engine 90 generates the signature 120 by hashing the hash key 118 with the decrypted data 114.

At step 316, the generated signature 120 is compared to the received signature 56 to verify and authenticate the received data. At decisional step 318, a determination is made whether the signature 120 matches the signature 56. If the signature 120 matches the signature 56, the method ends. If the signature 120 does not match the signature 56, the method proceeds from step 318 to step 320, where the decrypted data 114 may be discarded. Additionally, an alert indicating that the signature 120 does not match the signature 56 may be generated at step 322.

Thus, the present invention provides secure data transmission without requiring certificates or other third party-provided information. Accordingly, the present invention substantially reduces or eliminates the likelihood of third-party interception and subversion of the transmitted data. Additionally, because the present invention does not encompass the use of time-based certificates, reliance on system clocks and the blind acceptance of potentially invalid certificates is substantially eliminated. Further, unlike secure shell or other tunneling protocols, the encryption key changes with each transmitted data packet, thereby further reducing the likelihood of third party interception and subversion.

It should be understood that in the described methods, certain steps may be omitted or accomplished in a sequence different from that depicted in FIGURES 2 and 3. For example, referring to FIGURE 2, step 208 of generating the character string 54 may be accomplished at any time prior to the step 210 of generating the hash key 64. Also, it should be understood that the methods depicted in FIGURES 2 and 3